Ms. No.: CVIU-12-237
Title: SnooperText: A Text Detection System for Automatic Indexing of Urban Scenes
Authors: R. Minetto, N. Thome, M. Cord, N. J. Leite, J. Stolfi
Replies to Reviewers' Comments of 2012-12-07

1 General comments

Editor: *The reviewers found that the article was not acceptable in its present form. However, if you feel that you can suitably address the concerns and issues raised by the reviewers in their comments below, I would welcome receiving a revised manuscript.*

> We have substantially revised the paper taking into account the Reviewers' comments, which were quite helpful. Please reconsider the paper for publication in CVIU.

Reviewer #1: *I think that the paper does not provide evidence sufficient for making the readers convinced.*

> We hope that the new information included in the revised paper, together with the paper on the T-HOG descriptor we recently published [**?**], will be sufficient for the reader to properly evaluate the SnooperText system.

2 Topic and scope

Reviewer #1: *The paper proposes a text detection method based on [4] (?) to combine it with an existing free OCR system, Tesseract.*

> OK.

Reviewer #2: *This paper reports a complete scene text detection and recognition system. The method consists of three main steps: multi-scale letter detection, letter grouping, text line verification (classification). The detailed techniques: morphology-based image segmentation, geometric filtering, letter classification by SVM, geometric grouping, text line classification using HOG feature, are not new, but are implemented elaborately to achieve high performance. On text line or word location, recognition is performed by an open source recognizer TESSERACT. The overall detection and recognition performance were evaluated on three public datasets, and were shown to be superior to state-of-the-art methods.*

> OK.

Reviewer #1: *The experimental results show that the proposed method does not outperform recent methods while it is comparable.*

> We have added to the comparison section three state-of-the-art detectors reported recently, namely Pan et al. [**?**] (2011), Neumann et al. [**?**] (2012), and Yi et al. [**?**] (2012). We found that SnooperText is still comparable to those newer algorithms. We also include the newer Google Street View benchmark (`SVT`) provided by Wang et al. [**?**], and found that SnooperText clearly outperforms the text detector of Neumann et al. [**?**], which was published in 2012.

Reviewer #2: *The major contribution of this paper lies in the well desgined overall processing flow, the elaborate implementation of detailed techniques, and the reported high performance, though the consistuent techniques are not original. There are many artificial parameters in the processing steps, but were appropriately specified in a easily understable way.*

> OK.

Reviewer #3: *The authors have presented a text detection scheme in Urban Scenes, which can be used for indexing.*

> OK.

Reviewer #2: *Page 8, title of Sect 2.2, "text classification" means classification of texts. In this paper, it should be "text region classification".*

> We have taken care of this. The module is now called "region validation" and the process "text/non-text region classification". ⋆[**Check and rethink as needed.**]

3 Contributions

Reviewer #1: *I understand the purpose of the paper is actually text recognition while the paper title is concerning about text detection.*

> We hope to have clarified that the main contribution of our paper is indeed about text *detection*, namely our (SnooperText) algorithm; and *not* about text recognition (OCR). The OCR performance is being provided only to show that SnooperText is suitable as a front-end for a typical OCR algorithm and to its motivating application (the iTowns project). We have now moved this part of the evaluation to a separate section in order to make this point more clear.

Reviewer #1: *Furthermore, important comparison is missing; how the proposed method is different from [P4] is not described*
Reviewer #3: *The work presented by the authors is an extension of their previous work published in ICIP 2010 (Reference [P4]), and a significant improvement in the results for the ICDAR and iTown databases have been achieved. The present work seems to have a substantial overlap with the previous work (Reference [P4]). Hence, a short description highlighting the steps, which have helped to achieve the better performance compared to the previous work (Reference [P4]), is required to understand the contributions in a better way (this might also be summarised in the "Highlights" section of the paper).*

> Our conference paper [P4] gave only a very superficial overview of an older version of SnooperText. Since then we have extensively tested the HOG-based text/non-text classifier module, and optimized it for text lines, resulting in the T-HOG descriptor. We described this work in a separate journal article [**?**]. In this new article we decribe in detail the other parts of SnooperText (character segmentation, geometric filtering, character grouping, and multiscale processing) and report its performance with additional databases and the optimized T-HOG descriptor. Another substantial constribution of this article is the use of text detection in the iTowns project. ⋆[**Provided that section grows to include interesting and quantitative information, such as the success rate?**] ⋆[**What else is worth saying about this.**]

Reviewer #2: *The T-HOG has no substantial difference from the original HOG. Do not say "novel" for it.*

OK. We have emphasized that the T-HOG is just an R-HOG with parameters optimized for text line recognition: chiefly a one-dimensional (1×7) cell array, smooth cell boundaries, and rescaling of the input image to a fixed height H preserving aspect ratio. $\star$[**Make sure we do this.**]

4 Bibliography

Reviewer #1: *If my understanding is correct, there are some missing references:*

[N1] A. Mishra, K. Alahari, and C. V. Jawahar, Top-Down and Bottom-up Cues for Scene Text Recognition, Proc. CVPR2012, 2012.

[N2] K. Wang, B. Babenko, S. Belongie, End-to-end Scene Text Recognition, Proc. ICCV2011, 2011

[N3] L. Neumann, J. Matas, Real-Time Scene Text Localization and Recognition, Proc. CVPR2012, 2012

We added the references and discussed them in the Related Work section. $\star$[**make sure we do this!**]

Reviewer #2: *The review of previous works in Sect 2 is well organized, but missed some important works published recently. I just mention two papers below, which also combine bottom-up component segmentation and top-down component/string verfication and report superior performance on public datasets.*

[Na] Y.-F. Pan, X. Hou, C.-L. Liu, A Hybrid Approach to Detect and Localize Texts in Natural Scene Images, 20(3): 800-813, 2011.

[Nb] C. Yi, Y. Tian, Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification, IEEE Trans. Image Processing, 21(9): 4256-4268, 2012.

We have included these references in the review of previous work. $\star$[**Do it!**]

5 Methodology and technical content

Reviewer #1: *The proposed text detection method is quite straightforward; it employs a basic image segmentation method followed by a new HOG-like feature and multi-resolution window scanning. Since the HOG feature is a traditional feature in the character recognition field, known as directional feature, I do not think they technically sound.*

> We hope that the revised article makes it clear that SnooperText is *not* a "window scanning" detector. Rather, it first locates individual candidate letters by segmentation and letter/non-letter shape classification, groups those candidate letters into candidate text regions by geometric proximity and alignment criteria, and only then applies the T-HOG text/non-text classifier to each text's bounding box, as a validation step. Moreover, the T-HOG is used in a way specific to text lines rather than character, namely with the image divided into 7 horizontal stripes instead of the usual array of cells. (A window scanning detector is briefly described in section ??? of the T-HOG paper, but only as an illustration of possible uses of the descriptor, and is not relevant to SnooperText.)

Reviewer #1: *The reason why a free OCR is employed in the work is not explained well.*

> We hope that this objection no longer applies now that we have clarified that the emphasis of the paper is text detection rather than extraction. ⋆[**The iTowns people should answer this one.**]

Reviewer #1: *If the purpose of the paper is really text recognition, I wonder why classification by SVM with HOG-like feature is needed for text detection. More concretely, I am wondering if it is possible to perform more efficient processing by replacing the class labels for text detection (text/non-text) with the class labels for text recognition (e.g., 52 alphabetic characters).*

> We take this comment as asking whether it makes sense to detect a text region before (or independently) recognizing the characters in it. From the current literature, this still seems to be an open question. In SnooperText, the T-HOGbased text/non-text validation module definitely improves the detection F score as well as the end-to-end

scores F' and F''. Theoretically, it seems likely that a text-containing region can be successfully classified as such even when the characters are too small to be identified with reasonable accuracy. In an application like iTowns, such a detector could be useful to alert the system's operators about the presence of text that may deserve to be visually deciphered and entered by hand.

Reviewer #3: *More details on the text-line descriptor which has 27-63 features, is highly desirable. Figures with clear descriptions would be useful.*

These details are now available in a separate paper [**?**].

Reviewer #2: *Page 9, line 9, 8 cells with 8-bin HOG plus 1 mean difference and 1 sd should give 66 features, which contradict with the descriptor of 80 features.*

Indeed. We have corrected that line. SnooperText actually uses a T-HOG descriptor with 63 features (7 histograms with 9 bins each, without any extra features).

Reviewer #2: *Page 12, the paragraph above Fig 9, it is not clear how "isolated letters are discarded." Are there any thresholds for the letter size?*

We have clarified this point. The candidate characters are filtered for minimum and maximum size before being given to the character grouping module. In the latter, any candidate character that could not be grouped with other characters is discarded. ⋆[**Check!**]

Reviewer #2: *Sect 4.1, only the test datasets wre described. What about the training data (for training letter and text line classifiers)?*

⋆

[**See Reviewer #1**]

Reviewer #3: *The system presented is a heuristic-based method and many thresholds have been used. The performance of the system seems to depend on the thresholds given for each respective dataset. Hence, in one sense, the system seems to be semi-automatic. On this basis [*with parameter tuning*], the authors have tested their system on three ground-truth datasets, and achieved competitive accuracy compared to the state-of-the-art systems available.*

> We have added a table showing all the parameters and their values as used for each benchmark. There are not that many parameters, and we have found that most of them can be set to "typical" values without significantly impairing performance. Adjustments seem to be needed only when there are substantial changes in the goals or in the nature of the images (e.g. when between scanned text vs. urban scenes, clean vs, noisy images, word vs. line detection). For a well-characterized image collection (such as the iTowns mosaics), we assume that the parameter values will be chosen at the beginning and then used without further change for the entire project.

Reviewer #3: *In sub-section 3.1.3 for letter/non-letter classification, four SVM classifiers were used. The SVM classifiers where trained using three image shape descriptors. Details about the training samples and how they were collected do not seem to be mentioned.*

⋆

> [**See Reviewer #1?**]

Reviewer #3: *The T-HOG descriptor proposed by the authors seems to be interesting. More details on training the SVM classifier, which was used for text/non-text classification, are required to fully understand the usage of T-HOG. Information such as, what are the text and non-text samples used for training the SVM classifier, and how they were collected, would be helpful.*

> We have clarified the training of the T-HOG-based region validation module.

Reviewer #3: *The results of geometric filtering appear to be for a region rather than the individual character/letter; it is not entirely clear how the classification was performed. A clear illustration using figures would be highly desirable.*

> We revised the text and hopefully its now clear that the geometric filtering is based on the dimensions of the segment's bounding box. ⋆[**The area, height, width and aspect ratio criteria may be redundant. Check! A figure showing all valid (w, h) pairs may be helpful.**].

6 Testing and metrics

Reviewer #1: *The proposed method should be compared with [N1,N2,N3]. While I cannot do direct comparison since experimental conditions are different, I guess it is not difficult to perform experiments [*for the first two*] in the same condition.*

> We have included the text detection performance data of the detector of Neumann et al., and that of the of Wang et al. as reported by Neumann et al.. ⋆[**Make sure we did this.**] We also included the end-to-end scores reported these three papers in the OCR performance section.Unfortunately, as the reviewer notes, the Mishra et al. paper reports only the end-to-end performance, including the final OCR step, so we could not compare the performance of their detection module with that of SnooperText.

Reviewer #2: *Table 1, I suggest to add the results of the above papers [Na][Nb]*

> We have included their performance data in the comparison with SnooperText.

Reviewer #1: *[*The present method and the one in [P4]*] are not compared in the experiments.*

> Since the system superficially described in [P4] is only an an older version of SnooperText, it is no longer of interest. ⋆[**Were there performance numbers in [P4]?**]

Reviewer #1: *In Sec. 4.2.1, there is a sentence that "We note that the first method [*for averaging scores*] suffers from higher sampling noise and a negative bias compared to the other two." This needs a more detailed explanation.*

> We have clarified this point in the section *Rectangle-based performance metrics.*

Reviewer #1: *In Table 3, I found a number '0.24' exceeds the ideal value '0.10' in the row of* TessFront+T-HOG *and column of P' in* OCR scores (rigorous)*. A similar thing happens also in Table 1. While I understand they can happen, they indicate that the definition of "ideal" is not correct. I think the ideal values should not appear in the column of P', R', P'' and R''.*

We replaced the name ‘Ideal’ by ‘Cropped’, wich seems to be the name commonly used for this “detector” in the literature [**?**, **?**]. ⋆[**Do that!**] (Needless to say, one may obtain a higher end-to-end precision P' or P'' with an imperfect text detector than with the perfect ‘Cropped’ detector, if the former preferably loses words that the OCR back-end would mis-read.)

Reviewer #3: *The authors could have also included another section for analysis of the failure cases with some sample images, which describe the probable reasons for failure. This would help the reader to understand the drawbacks/limitations of the proposed system.*

⋆

[**Check whether we have done enough on that.**]

Reviewer #3: *Adding some details on time complexity of the proposed method would also be useful.*

⋆

[**Consider this.**]

7 Format and style

Reviewer #1: *I found many mistakes in English. Especially, some sentences are not real sentences.*
Reviewer #2: *The overall structure and language presentation of the paper are acceptable.*
Reviewer #3: *Correcting typos will enhance the quality of the paper.*

We have rewritten subtantial parts of the article to improve its grammar, style, and clarity. ⋆[**Run a spell checker and re-read carefully.**]

Reviewer #1: *For example, I feel the first sentence in Sec.1 is strange. Is it broken?*

Indeed. It’s fixed.

Reviewer #1: *Sec. 3.1.1 is hard to understand. The first sentence in Sec. 3.1.1 is bit strange. The first sentence in the second paragraph in Sec. 3.1.1 beginning with "In order to segment the image, first it is computed a local background image B" is also strange.*

> OK, fixed.

Reviewer #1: *I is not well defined*

> OK, fixed.

Reviewer #1: *how to create B and F are not written.*

> We have added a reference to a Matheatical Morphology source that defines grayscale erosion. ⋆[**Explain that it is a square element**]. ⋆[**Why square? Round should be better. Fuzzy even better.**]

Reviewer #2: *Page 12, 3rd line from bottom, "three-vector" should be "three-dimensional vector."*

> OK. We changed to "three-element vector." ⋆[**Check!**]

Reviewer #2: *Page 15, 3rd line, "the the" should be one "the"*

> OK, fixed.

Reviewer #2: *Title of Fig. 12, "character recognition" should be "character detection." In the paragraph below Fig. 9, 3rd line, "recognition" is also "detection." There maybe other places in the paper that confuse "recognition" and "detection," please check carefully. "Text line classification" is better be "Text line verification."*

> Indeed. We have thoroughly revised the paper in this regard.

References